AN ICONIC USER INTERFACE FOR GEOMETRIC MODELLING

José Carlos TEIXEIRA *
Joachim RIX

Fraunhofer-Arbeitsgruppe
Graphische Datenverarbeitung
Wilhelminenstraße 7
D - 6100 Darmstadt
Tel. (49) 6151 1000-26/25

ABSTRACT

The User Interface plays a very important role in the geometric modelling systems. Different approaches have been developed in the last years, with and without the use of graphic images. As the sight is the most important human sensory, more and more efforts have been made to design effective and powerful graphical oriented User Interfaces.

Starting with general issues and guidelines for graphical User Interface design, the paper reports the DACAD User Interface approach. DACAD is an experimental system that uses the methods of the constructive geometry and the technical design knowledge as a basis to create and manipulate three dimensional models.

Keywords : *User Interface, dialogue, computer aided design, modelling.*

* *Grupo de Métodos e Sistemas Gráficos*
Dep. de Matemática - Faculdade de Ciencias e Tecnologia
Universidade de Coimbra - 3000 Coimbra - PORTUGAL

1. INTRODUCTION

When someone designs and implements a computer system, he also has to build a User Interface. It is very well known that an unlucky User Interface can negatively influence the use of an excellent programme and, on the other hand, a suggestive and attractive User Interface hides some deficiencies of a programme being not as powerful. I. e., the success of a computer system is directly related to the quality of his User Interface.

As primary issues of a User Interface design we consider the simplicity to use and the self-explanation of his concepts. In fact, the user should not be distracted from his main task, i. e., certainly not to know how to work with the User Interface. An important goal will be achieved if the system "disappears" from the user's consciousness and therefore he or she concentrates his/her energy on the problem to solve or task to execute. A user wishes a service from the system, not to serve the system!

The end user must always be in the system designer's mind. That is a very important issue, because a system is not designed for its designers who, normally, are not the end users. In this context it is important to emphasize the user's diversity and not to forget that one important goal in the User Interface design is to allow people using the system who are not computer professionals.

As said Salmon [SARS-87] "*the users are always the starting point for a good human-computer interface*". If a User Interface provides a transparent mean close to the user's mental model, reduces the user's effort to learn and remember the system commands, and simplifies and provides a rapid feedback to the user actions, it fulfils the most important goal: the user satisfaction. But how to achieve this goal?

A lot of User Interfaces have been designed based on intuition. The R & D work in the User Interface area is too new (20-30 years) and so this situation can be seen as a normal one. Nowadays a lot of efforts have been made to build a "theory" and to make the User Interface design a science. In this context the interdisciplinarity character of the User Interface design assumes particular importance, especially the important input from the human factors and psychological research. For instance, it is very important to evaluate the user's adaptability and to know how to minimize the stress and therefore to increase the productivity. All of this basic research provides, at least, a way of thinking and guidelines in the User Interface design.

One of the most important guidelines in the User Interface design is the involvement of computer graphics specialists in the design and implementation of User Interfaces. That means the increasing use of the graphics capabilities, especially in the User Interfaces of CAD systems. As said J. Foley [FOJW-84] "*The promise of interactive graphics is to provide a user-computer communication medium that is at once benign, responsive and graphic*".

After 1981 one important step in this direction was the spread of the basic ideas developed over the previous ten years at the Xerox Palo Alto Research Centre (PARC) through the STAR User Interface. In spite of the STAR's application area - desktop publishing - the ideas have spread rapidly and have been applied in an increasing number of systems. Some of the principal ideas were: the illusion of manipulating objects, visual order and user focus, consistent and appropriate graphic vocabulary and matching with the medium. The WYSIWYG (What You See Is What You Get) philosophy is well known !

For this development the continuous decreasing graphics hardware prices have played a very important role, allied to its increasing functionalities, especially in the following technologies: computing power, raster graphics and pointing devices. So it is possible to increase the availability of good hardware and the satisfaction and ease to use it.

In our work we consider especially aspects related to graphical User Interfaces, an area of great innovation. The graphic communication constitutes one of the richer and clearer communication means, and this should be explored.

2. GUIDELINES OF GRAPHICAL USER INTERFACE DESIGN

Blake [BLAT-87], mentioning Norman [NORD-83], enumerates some principles for User Interface design:

- *there are no simple answers, only tradeoffs*
- *interface design should be a separate activity*
- *low-level protocols are critical*
- *information retrieval dominates activity*
- *activities (operations) are structured in a task-specific way*

These principles, we prefer to call them guidelines, are some of those we can easily find in references of different authors. Most of them are the result of the experiences with system's implementation. First we will briefly enumerate some commonly accepted guidelines for User Interface design.

2.1 - Role of Users and Computers

Before starting a User Interface design the designer must be aware of the tasks that should be addressed to the user and to the computer. According R. Pew [PEWR-83] the users are good in the following activities:

- monitoring of the systems
- locating and recognizing patterns
- versatility in handling different input and output symbols
- ability to adapt to risk and uncertain situations

The computer can play a good role in the following activities:

- tasks requiring large amounts of memory capacity
- deductions where the information and rules are complete
- repetitive and time consuming tasks

The correct addressability of tasks can be the beginning of success!

2.2 Screen layout

The screen layout should improve, as well as possible, the normal user's way of work. As important issues we can refer to those related to the general organization and the space for different tasks:

- simple organization
- spaciousness for the application tasks

- asthetics
- reduction of the eye movements
- consistence in the different locations, especially in the related ones
- logical grouping of items
- highlighting the information according the status of the system and the needs for next steps
- limiting the information available to the user: it must only be the one needed for a task
- optimizing the space free for graphical constructions and other operations over graphics representations
- the same type of information, graphical or textual, should appear always at the same place

2.3 Comprehensibility of the User Interface

If we accept that the most successful User Interfaces are those which incorporate graphics, it is easy to understand that it is not only important to choose appropriately the objects manipulated in one application, but also their visible representation in the system. Each of them must be familiar to the user and rapidly identified and distinguished from another ones. That means, the choice of the icon's set for the objects and operations over them is of capital importance. We use the word ICON to denote, as Dave Smith [SMID-77], a sign, symbol or pictogram. The quality of an icon is a function of the wanted object representation and other characteristics of the icon itself: shape, size, colour of object representation and background, etc. An icon of good quality stands by itself and does not need a label.

Another important point to denote here is the capacity of the User Interface to capture the user's attention. An expressive User Interface helps the user to find always the important task or information.

2.4 Consistence of the graphic vocabulary

Related to the preceding point, the importance of the graphic vocabulary used should be emphasized, especially the following points:

- All the icons representing objects of the same class must have the same graphic characteristics.
- The different functionalities and status of the system can be emphasized by the icon's appearance, but must follow the same general principle.
- The icons must be grouped appropriately and always with the same philosophy.
- The highlight of active information and options must be the same along the whole system.
- The graphical information available at each moment, especially for the dialogue, should follow the same principle.
- The system messages should be concise, clear and with the same format.
- Different cursor shapes can be used to indicate special modes.

2.5 Affinity with the hardware capabilities

The quality of the graphic User Interface must be appropriate to the market's demands. So special attention is needed when choosing the hardware. For instance a large display with high resolution should be chosen. On the other hand, the User Interface designer should explore the characteristics of a given equipment and adapt the solutions to them. It is important to be aware of the fact that the solutions with a monochromatic display can be different from the solutions for a colour display.

2.6 Interaction style

The success of a User Interface depends on the user's view of the information presented by the system, how he or she executes the actions and how the system reacts to these actions. In the graphical oriented User Interfaces all the interaction process is simplified. If we compare the graphical oriented User Interfaces (GOUI) with the command-driven textual interfaces (CDTI) the benefits of the first ones are evident:

- In the CDTI the user has to remember commands, its syntax and applicability.
- In the GOUI the memorization is drastically reduced because the representations of the objects, as well as the possible operations, are presented on the screen.
- In the CDTI the user must type the commands with the correct syntax.
- In the GOUI the user has only to select by pointing.
- In the GOUI the user interacts continually with representations of the objects.

We can also refer to the fact that the graphical oriented User Interfaces are well adapted for the WYSIWYG philosophy, and mention Dave Smith [SMDI-82]:

> *"A subtle thing happens when everything is visible: the display becomes a reality. The user model becomes identical with what is on the screen. Objects can be understood purely in terms of their visible characteristics. Actions can be understood in terms of their effects on the screen. This lets users conduct experiments to test, verify and expand their understanding - the essence of experimental science"*

3. THE DACAD DESIGN APPROACH

In design of the DACAD User Interface we have basically considered the methodology proposed by James Foley [FOJW-84], also referred to by Rod Salmon [SARS-87] and with the changes introduced by J. Foley [FOLJ-87]. Based on this related work, we have used a top-down methodology with the following tasks:

- Definition of the main goals
- Type of End User characterization
- Conceptual design
- Functional design
- Dialogue design
- Lexical design
- Prototyping and Evaluation

Next we will report the basic options considered and some of the results obtained. All of this will be related to the preceding top-down design tasks.

3.1 Definition of the main goals

In the main goals definition the aims of the system to which the User Interface is needed should be considered. The main goals of the User Interface emerge from that ones.

DACAD intends to use the knowledge from the constructive geometry as a basis for geometric modelling [TEIJ-88]. The first goal is to provide an environment especially adapted to computer naive designers and engineers. The second goal is to have a basic system adapted to the education in constructive geometry and graphic methods, especially at the university level. The third goal is to obtain a prototype capable to be integrated in industrial applications.

3.2 Type of End User characterization

It is well known that it is difficult to determine exactly the set of a system's end users and to estimate its diversity. As principal possible end user types should be considered: application naive, application experts, computer naive, programmers, system experts and marketing personnel. Functionalities at the User Interface level required for the naive users and marketing personal can be unnecessary for system experts. So it is very important to define the main set of end users for which the system, and therefore the User Interface, will be designed.

From the main goals defined before, it is clear that our main set of end users are people with knowledge in geometry, classical graphic methods and graphical design techniques and application experts, but not necessarily computer experts. The community of the computer experts and programmers is the last priority in our work.

3.3 Conceptual design

The conceptual design processes at an abstract level. The user's job must be understood and observed what actually the user does and how. It is also important to foresee the changes in the user's work, the learning time needed and the improvement in the user's satisfaction.

The conceptual model defines also the set of functionalities provided by the system, therefore by the User Interface, and the types of objects manipulated, with the respective relations and attributes.

DACAD allows the use of rules and constructions, simple and complex ones, which are well known for draftsmen and designers. The rules and constructions should allow the work on one plane, not necessary one of the orthographic planes, and provide the construction of three-dimensional models. The incorporation of the methods in use, an important goal of our work, should not be a constraint for new ones. However, special attention is needed to allow the user, in an easy and fast way, to recognize and use the available functionalities.

Related to the type of end users, the system does not use a programming language in its dialogue. The user can work with a minimal use of his memory, therefore the User Interface is transparent and contains sufficient information for the work and reduces the unnecessary one. This goal is achieved with a graphical oriented User Interface, which explores the functional characteristics of the users. In this way we also reduce the learning time, we increase the speed of use and, we hope, the user satisfaction.

In different situations the introduction of new CAD systems, positive and necessary, is an additional problem for some designers with a lot of experience. The cause for that situation can possibly be found in the development of some CAD systems, process oriented to the system's problems instead of the user's ones. On the other hand, we think that students need to work with the methods of the descriptive geometry, and so of the constructive geometry, in order to work better with or without CAD systems. So this work orientation seems to have also appeal justification.

In the first version of DACAD we consider three classes of objects:

Constructive_objects - three-dimensional objects through which a three-dimensional model can be described. These objects are defined from projections on planes.

(*Line_segment, Polyline, Polygon, Circle, Circular_arc, Ellipse, Elliptic_arç, Prism, Cylinder, Cone, Sphere*)

Primitive_objects - two dimensional objects created and manipulated on a plane.

(*Point, Line, Line_segment, Polyline, Polygon, Circle, Circular_arc, Ellipse, Elliptic_arc*)

Help_objects - primitive_objects that cannot be constructive_objects :

(*Point, Line, Half_line*)

For each primitive_object exists a set of definition methods that can be executed on one plane. Each definition method is a terminal process that orients and controls the user's work. We think that is one way to help the user, to prevent errors and to ensure the consistence of results.

To obtain a constructive_object, a complete_definition method must be executed. Each of these methods solves the problem: from which constructive_object is a primitive_object a projection.

Primitive_objects like polygon have a hierarchical structure: it is possible to identify and manipulate the polygon, each of its line_segments or each of its vertices.

The possible object graphical attributes are those provided by the graphic system used, like marker type, colour and style.

3.4 Functional design

The functional design takes the conceptual design as a basis and defines, first in an abstract level, the meaning of the user's procedures. The system's functionalities are detailed, in particular the information needed for each operation, the possible semantic errors and the result of each operation. What kind of graphical information and how it should be displayed, it is another point to define. The sequences of inputs and outputs, the "grammar" of the dialogue and its elementar tokens should also be defined.

Each definition method can be described as a sequence of user and system actions. For this description we have developed a set of syntax rules [TEIJ-88]. As an example we present here the description of a definition method:

Definition of a circular_arc tangent to one line, half_line or line_segment, from the radius, the tangential point on the line_item and the two extreme points or one extreme point and one central angle.

Circ_arc_tang_line_radius_point : SEQUENCE {

IDENT line_item (ln) ;
(* *ln is the reference line, half_line or line_segment* *)

IDENT point (pt) or DEF point (pt) ;
(* *pt is the tangential point on the line_item* ***ln*** *)

DEF radius (rad) ;
(* ***rad*** *is the radius of the new circular_arc* *)

G_AST circle (c) ;
(* c *is one assistance circle tangent to* **ln** at **pt**, *which radius is* **rad** *)

IDENT_R point (p1) or DEF_R point (p1) ;
(* ***p1*** *is one of the extreme points of the new circular_arc, on the assistance circle* ***c*** *)

CHOICE ((IDENT_R point (p2) or DEF_R point (p2)) ,
DEF angle (ang)) ;
(* ***p2*** *is the other extreme point of the new circular_arc, on the assistance circle. We can also define the central angle defined. by* ***p1*** *and* ***p2*** *: + or -* *)

[S_CONST point (p2)] ;
(* *This construction exists only if the user defines an angle* *)

S_CONST circular_arc (circ_arc) ;
(* ***circ_arc*** *is the new circular_arc* *)

CHOICE (S_CONST delete part of the referenced line_item ,
S_CONST only new circular_arc) ;
(* *after the circular_arc's construction two situations are possible :*
- the line_item is modified (one part is deleted)
- no modifications occur *)

[S_CONST line_item (ln1)] ;
(* ***ln1*** *is the new line_item, if the user had choosed the first preceding choice* *)

}.

INPUT_OBJECTS : - line_item (ln)
- [point (pt)]

```
OUTPUT_OBJECTS : - circular_arc (circ_arc)
                 - point (p1)
                 - point (p2)
                 - point (pt)
                 - [ line_item (ln1) ]
```

Each definition method is a terminal process so, if a user executes a definition method only the dialogue information related with itself is active. If an error occurs, a textual prompt is presented to the user. It is always possible to cancel the execution of a method or undo its sequential actions.

For the execution of a definition or a complete_definition method the user must have at least one projection plane on the screen. The default situation is the presence of the three orthographic planes and a fourth plane for an axonometric or perspective projection. The user can also define a general plane and work on it.

The user also has always visible information about the input method used, scale and unity chosen, name of the active model and name of the visible planes.

3.5 Dialogue design

At this step the options related to the dialogue style should be assumed, i. e., the designer should define if the user's dialogue should be processed, for instance, in a Question-Answer mode, with a Command Language or with Menu Selection. The issues from the last three steps are very important in this decision, as well as the end user's factors: learning time, satisfaction, time to get skill and speed on use.

To decrease the training time and to facilitate the use, we selected as main interaction style the menu selection by pointing. In the menu selection there are some default selections, which are, like the actual ones, explicitly represented by a raster, to help inexperienced users.

We have the following types of menus:

`Main functionalities` - menu that groups the main functionalities of the system, like Edit, Complete, Delete, Insert and Manipulate.

`Input` - menu with the possible means to input data: from a cursor driver (e.g. mouse) and from a direct input device (e.g. keyboard)

`Graphical presentation`
- *Projection plane present* - menu with the possible planes present on the screen: ZOX, XOY, YOZ, axonometric/perspective, OTHER and ALL (the first four)
- *Graphical attributes*
  - *Marker attributes*
  - *Line attributes*

Application functionalities

- *Primitive objects* - menu with the primitive_objects
- *Definition methods* - for each primitive_object a menu that groups its possible definition methods
- *Definition method execution* - for each definition method a menu which groups the sequence of actions needed for the execution of the method
- *Complete object* - menus for the different phases of each complete definition method

Each of these menus is enclosed with a box and not labelled.

3.6 Lexical design

After the preceding steps the designer is able to implement the concrete form of the User Interface. The hardware devices needed for the input and output are chosen and the hardware and basic software capabilities are explored. The output primitives needed and their attributes should be identified. The User Interface's layout is also of capital importance .

Between the available possibilities, the choice of equipment for the implementation was based on: good graphic capabilities, good computing power and basic software. With regard to the first point, we use a large display with a resolution of 1280x1024 pixels.

First of all we had designed the User Interface's layout based on the guidelines referred to at 2.3. After an iterative process we achieve the actual version shown in Fig. 1.

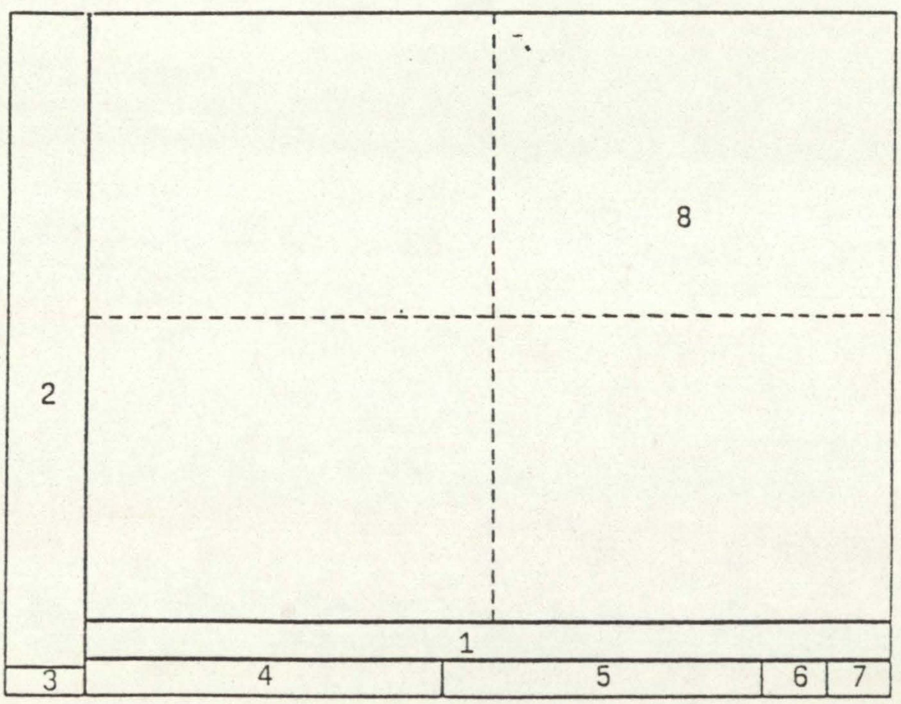

Fig. 1 - DACAD layout

The layout of DACAD presented in Fig. 1 consists basically on 8 areas:

1 - for work options and system commands menus

2 - for status and work menus

3 - for the display of the name of the model

4 - for prompts and error messages

5 - for input of data

6 - for the display of the cursor position

7 - for the display of unity of work, scale and raster interval

8 - for graphic output and graphic manipulation

We have two types of icons: text_icons and graphic_icons. The use of text_icons is limited to the main functionalities of the system and to the CANCEL and UNDO functionalities. We believe that this is not contradictory with our main goals: first the set of text_icons is very small compared with the set of all icons; second for functionalities like EDIT and DELETE with a clear semantic, the use of a graphical symbol could not be more effective. All the other icons are graphic_icons. Some attention was taken to the shape of the different icons, related to their semantic. We use different shapes to: icons with the main functionalities of the system, object_related icons and graphical attributes icons. In Fig. 2 an example of each one.

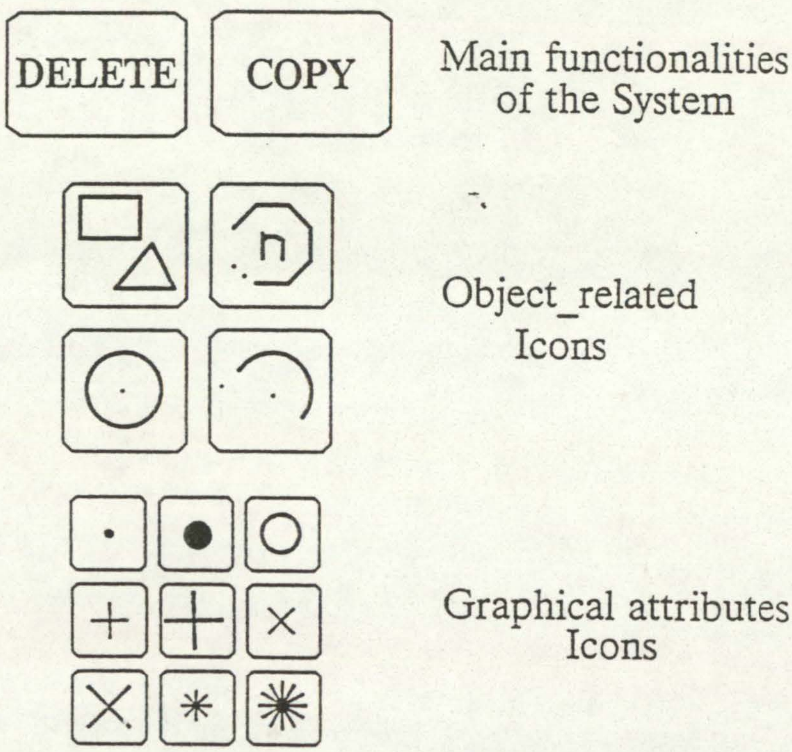

Fig. 2 - Different types of icons

To reinforce the presence of a definition method menu, the first icon, which represents the actual definition method, is boxed with thick lines. The sequential actions are vertically ordered and the possible choices horizontally, that means, the presence of two icons on the same horizontal indicates a choice for the user. In the last row of each definition method menu the user has the CANCEL and UNDO functionalities. In Fig. 3 a menu for the definition of a triangle is presented. We can also refer to the first icon's always being rastered, and the raster on another icon indicates the actual step on the method execution.

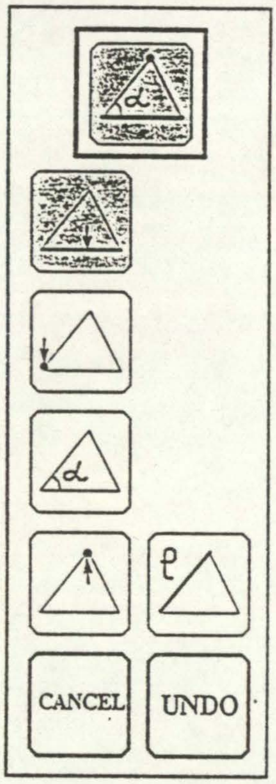

Fig. 3 - Definition of a triangle

Another point was the cursor design. We have designed the cursor presented in Fig. 4 because we think, that it has at least some of the required qualities: easy to move through the display, to highlight the pointing functionally and not to obscure the screen information.

Fig. 4 - Cursor of DACAD

3.7 Prototype and Evaluation

The design is an iterative work. Therefore the implementation of a User Interface's prototype can be of special importance for the refinement and improvement of the first decisions. This is the only way to experiment alternatives and to get input from the potential users. The evaluation of all contributions allows one more based decision about the improvements needed. One barrier to a good User Interface design is the belief that the best solution is the first one!

A first prototype was realized and evaluated for two kinds of people: User Interface designer experts from the Darmstadt Computer Graphics Centre and students from the Darmstadt Technical University. From the first group we want to thank the colleagues Wolfgang Hübner and Jairo Cote Muñoz for their valuable criticism. The present version is the result of some improvements after the first evaluations. We intend to evaluate our prototype also with professional designers.

4. IMPLEMENTATION

DACAD has been implemented on SIEMENS WS 30 workstations (similar to the APOLLO DOMAIN 3000) running under an UNIX operating system. The whole User Interface was implemented with the resources provided by a graphical kernel system. A library of tools was implemented for the functionalities required.

5. CONCLUSIONS

DACAD should be seen as a prototype from R & D work. The main results of its development are:

- ♦ a contribution to the incorporation of the constructive geometry methodologies in geometric modelling
- ♦ a contribution to a graphical oriented work methodology
- ♦ the improvement of the user_interface to allow the preceding goal

For the future we plan:

- ♦ to extend the work methodology to other geometric modelling techniques, like the modelling of free form surfaces
- ♦ to extend the object sets to allow product oriented modelling techniques
- ♦ to relate our User Interface approach with the standard activities in this area

6. REFERENCES

[BIHM-88] - **Bittner, H., Muñoz, J. Cote, Eser, F. and Franz, D.**, " *A user interface management system for integrating electronical and mechanical CAD* ", in: Proceedings of the IFAC Conference Man-Machine Systems, Oulu, Finland, June 1988, pp. 505 - 510.

[BLAT-87] - **Blake, Tyler**, " *Introduction to the art and science of User Interface Design* ", in: CHI + GI 1987 - Human Factors in Computing Systems and Graphics Interface, April 1987, Toronto, Canada.

[ENJS-88] - **Encarnaçao, J. and Schönhut, J.**, " *High Performance, Visualisation and Integration - The Computer Graphics Headlines for the 90's* ", in: Proceedings of the IFIP TC5 Conference on CAD/CAM Technology Transfer to Latin America, Mexico City, August 22 to 26 1988, pp. 97-111.

[FOJV-82] - **Foley, James D. and Van Dam, Andries**, " *Fundamentals of Interactive Computer Graphics* ", Addison-Wesley Publishing Company, 1982.

[FOJW-84] - **Foley, James D., Wallace, Victor L. and Chan, Peggy**, " *The Human Factors of Computer Graphics Interaction Techniques* ", IEEE CG&A, November 1984, pp. 13-48.

[FOLJ-87] - **Foley, James**, " *Models and tools for the designers of user-computer interfaces* ", Report GWU-IIST-87-03 Department of EE & CS, George Washington University, Washington. D.C. 20052, 1987.

[HÜWL-87] - **Hübner, W. , Lux-Mülders, G. and Muth, M.**, " THESEUS, *die Benutzungsoberfläche der UNIBASE* ", Springer-Verlag, 1987.

[NORD-83] - **Norman, D. A.**, " *Steps Toward a Cognitive Engineering: Design rules on analyses of human error* ", Communications of the ACM, Vol. 4, 1983, pp. 254-258.

[PEWR-83] - **Pew, R. W.**, " *Human Skills and their utilization* ", Lecture Notes, Man-Machine Interface Symposium ,CEI Europe, June 1983.

[SARS-87] - **Salmon, Rod and Slater, Mel** " *Computer Graphics - Systems & Concepts* ", Addison-Wesley Publishing Company, 1987.

[SHNB-87] - **Shneiderman, B.**, " *Designing the User Interface: Strategies for Effective Human-Computer Interaction* ", Addison-Wesley, 1987.

[SMDI-82] - **Smith, D. C., Irby, C., Kimball, R. and Verplank, B.**, " *Designing the Star User Interface* ", in: Byte, April 1982, pp. 242-282.

[SMID-77] - **Smith, Dave C.**, " *Pygmation, A Computer Program to Model and Simulate Creative Thought* ", Birkhäuser Verlag, Basel and Stuttgart, 1977.

[TEIJ-88] - **Teixeira, José C.**, " *Interface with geometric modelling systems based on constructive geometry* ", in: Proceedings of the IFIP TC5 Conference on CAD/CAM Technology Transfer to Latin America, Mexico City, August 22 to 26 1988, pp. 159-170.